REMARKS

Claims 11-18 and 20-22 are pending in the application. The Examiner has rejected Claims 18 and 20-22 as being directed to non-statutory subject matter. By this amendment, Applicants are submitting amendments to Claims 18 and 20-22. Applicants believe that the amendments address the Examiner's concerns and respectfully request withdrawal of the rejections.

The Examiner has rejected Claims 11-13 and 18-19 under 35 USC 102(b) as anticipated by Chess; and, has rejected Claims 14-17, and 20-22 under 35 USC 103 as being unpatentable over Chess. For the reasons set forth below, Applicants believe that the claims are patentable over the cited art.

The presently-claimed invention provides a method and apparatus for a computer for enabling a user to provide input values to a running program after the program has begun running by prior to the program requesting those input values. The method steps, as recited in Claim 11, include maintaining a bag buffer of variable/value pairs in the program, wherein user input values are substituted for program variables during program execution, receiving a communication, including input values, from the user, and temporarily storing the input values in the bag buffer until those value are need by the program. Similarly, the structure as recited in Claim 18 comprises an output buffer for storing output values to be displayed to a user; a bag buffer for storing variable/value pairs for use by the program; an input buffer for storing values for which user input of variables is required; and a program state buffer for storing at least the present state of the program.

The Chess article is directed to the use of itinerant agents for mobile computing. The itinerant agents are described as "programs, dispatched from a source computer, that roam among a set of networked servers until they accomplish their task." Under the Chess teachings, an itinerant agent is initialized with a user's task and is dispatched to accomplish the task. When creating a task for the itinerant agent, the user employs a form or dialogue to input the task specification, which is then converted into a transaction agent program capable of executing the task. All user input is conducted prior to running of the program. The Chess article does not specify how user input is stored. Further, the Chess article does not teach whether user input, such as user preferences, is used for program execution. Applicants disagree with the Examiner's interpretation of the Chess teachings.

With regard to the language of the method claims, Claims 11-13, Applicants respectfully assert that the Chess article does not anticipate the invention as set forth in the independent claim. Claim 11 recites a method for enabling a user to provide input values as variables to a running program after said program has begun running and before the program needs the input values, wherein user input values are substituted for program variables during program execution, comprising the steps of maintaining a bag buffer of variable/value pairs for use in executing the program in the program; receiving a communication, including input values, from the user; and temporarily storing said input values for said variables as variable/value pairs in said bag buffer. Applicants contend that the Chess article does not teach or suggest providing input values as variables to a *running* program, wherein the user input values are substituted for variables during program execution. Chess provides all input to the itinerant agent prior to task execution, and in fact prior to instantiation of the itinerant agent. Clearly Chess does not anticipate enabling a user to provide input values to a running program.

The Examiner has cited the Chess statement from page 39 that "[t]he agent is initialized with the user's task". Applicants contend that if the agent is initialized with the user's tasks, then that user is not providing input values to a program ***during program execution, after it has begun running but before the program requires the values.*** Chess expressly states that the inputting of the user's information initializes the agent. In contrast, the present invention explicitly teaches and claims that the user input values are received during program execution but before the values are needed by the program.

The Examiner further cites the Chess teaching that "[h]e uses a form or a dialogue to state his need". Again, Chess has the user inputting information in order to initialize the agent (i.e., in order to start program execution). The user is not inputting values during program execution but before the program requires those values. If the Chess user does not input the values, the agent/program does not begin executing. Moreover, once the Chess agent is initialized, there is no mechanism taught for the Chess user to input values, other than when prompted by the agent (i.e., when the program needs the values).

Applicants further assert that Chess does not anticipate providing values for variables wherein the values will be substituted for variables during program execution. The cited Chess teachings simply states that the Transaction Agent is given the user's preferences (page 36, right column, last paragraph), but does not teach or suggest that those user preferences be used during task execution by the itinerant agent. While Chess says that the user's preferences are "expressed are rules", Applicants respectfully assert that Chess does not teach that the user's preferences are used as input values for program execution. Absent some express teachings, it cannot be maintained that Chess anticipates the claim language, which explicitly recites storing input values in variable/value pairs *for use in executing the program*.

Applicants further assert that Chess does not anticipate the claimed step of maintaining a bag buffer of variable/value pairs for use in executing the program in the program. As noted above, the Chess article does not provide any details of how user input information (e.g., the user preferences) are stored. The cited "goals and status information" from page 39, illustrated in Figure 2 of Chess, provides a very vague description of an agent's structure, but clearly does not teach or suggest storing variable/value pairs in a bag buffer, wherein the user input values are to be substituted for program variables during program execution.

Applicants further assert that the Chess article does not anticipate the claimed steps of receiving a communication during program execution, including input values, from the user and temporarily storing said input values for said variables as variable/value pairs in the bag buffer. Chess has the stated intention of providing a mechanism for an itinerant agent to receive user input at agent initialization and be dispatched without any further user input. There is nothing in the Chess article which either teaches or suggests providing user input during program execution.

Since the Chess article does not teach or suggest the maintaining of a bag buffer, the recieving input values from a user during program exeuction or the temporary storing of input values in the bag buffer, it cannot be maintained that Chess anticipates the language of independent Claims 11 and 18.

With respect to Claim 19, Applicants note that the Examiner has rejected it on page 5 as anticipated by Chess. However, Claim 19 has been canceled.

With regard to the structure claims, independent Claim 18 and those claims which depend therefrom and add further limitations thereto, Applicants assert that Chess does not provide any details for storage of data. Chess does not teach or suggest an output buffer, an input buffer, a program state buffer, and a bag buffer as claimed. Applicants reiterate that Chess does not store variable/value pairs of data, which data is needed for execution of the program. The stored variable/value pairs of the present invention are provided by the user and stored for use by the program while the program is running, but prior to when the program actually needs the variables/values. There is simply nothing in the cited Chess teachings which anticipates or obviates that claim language. In rejecting the claimed output buffer, the Examiner states that a "client sends its agent...to retrieve the latest version of a technical paper...[serving] as a courier...for data and program content." Applicants fail to see how that statement anticipates the claimed output buffer for storing program execution output values to be displayed to a user.

With respect to the claimed input buffer, the Examiner has cited the Chess teaching that "the agent is initialized with the user's task" and the passage on page 35 about the task specification. However, Chess does not teach or suggest an input buffer for storing values based on user input of values for variables required by an already running program, wherein user input values are substituted for program variables during program execution, said input buffer being accessed by said agent execution shell to communicate values for the input variables to the agent for present use by the agent during program execution. All that Chess states is that the user uses a form to "state his need". Such teachings clearly do not anticipate the claimed input buffer.

With regard to the program state buffer for storing at least the present state of said program, the Examiner has cited the Chess statement that "when the agent has successfully completed its task...it may collect its state." Chess does not, however, teach a program state buffer.

Finally, with regard to the claim feature of a bag buffer for storing variable/value pairs for later use by the agent in executing the program, Applicants reiterate the arguments presented above, that Chess does not teach how user preferences are stored, and clearly does not teach a bag buffer for storing variable/value pairs for use in executing the program. Applicants note that the Examiner cites the Chess statement that "the agent is initialized with the user's task" against the bag buffer. The Examiner has also cited the exact same language against the input buffer. Since Applicants are clearly reciting two distinct components, Applicants respectfully assert that one Chess teachings cannot anticipate two distinctly claimed components of the structure. The Examiner again cites the "goals and status information" which also does not anticipate a bag buffer for storing variable/value pairs.

It is well established under U. S. Patent Law that, for a reference to anticipate claim language under 35 USC 102, that reference must teach each and every claim feature. Since the Chess article does not teach a bag buffer, as part of a program, does not teach storing variable/value pairs in the bag buffer for use in executing the program, does not teach the user input of values during program execution but before the program needs the values, does not teach automatically accessing variables, and updating or disposing of input values, in response to a request for variables by the program, and does not teach a program state buffer in conjunction with input, output and bag buffers, it cannot be maintained that the Chess article anticipates the invention as claimed in Claims 11-13 and 18.

With specific reference to the language of Claim 12, Applicants disagree with the Examiner's conclusions. Chess' program is not retrieving or locating needed input values when the "agent...process[es] each of [the] candidates against the user's travel preferences". Chess's agent already has the user's travel preferences which were input to initialize the agent. What the agent is locating are candidates to evaluate against the user's travel preferences. Clearly the locating of candidates does not anticipate retrieving needed input values that have previously been stored.

With regard to Claim 13, Applicants again contend that ordering retrieved candidates is not the same as updating input variables. Further, releasing a hold on a candidate is not the same as or suggestive of putting variables in an input buffer for entry of values by the user. Finally, Applicants acknowledge that the Chess teaching of sending a page to a client is a form of notifying the client; however, that notifying is not done in conjunction with the other claimed steps. Accordingly, it cannot be concluded that Claim 13 is anticipated by Chess.

Claim 11 has also been rejected as anticipated by the Bull patent. The Examinr concludes that "explicit or implicit query parameters" of Bull anticipate the input values from Claim 11. Applicants respectfully disagree. Responses to a query/prompt by the system/program are not the same as or suggestive of providing input values to a program ***during program execution, after it has begun running but before the program requires the values.*** If a program is requesting query parameters,. it ***needs*** those values. With regard to the cited passage from Bull that "the user can establish a persistent...agent to monitor future information additions to the System", Applicants contend that instantiating a persistent agent to monitor changes to a system is not the same as or suggestive of a user inputting values that the program will later require. Monitoring changes to the system is not the same as storing values later to be needed by a running program. Applicants request reconsideration of the rejections as anticipated by Bull.

Applicants further assert that the Chess article does not obviate the invention as set forth in the pending claims. Applicants rely on the arguments set forth above with regard to the language of the independent claims. Further, Applicants respectfully assert that Chess does not teach or suggest the invention as set forth in dependent Claims 14-17 and 20-22. With regard to Claims 14-17, the Examiner has acknowledged that Chess does not expressly disclose notifying with the claimed electronic means yet has failed to cite any Chess teachings against the claim language. Rather, the Examiner states that "it would have been obvious to one skilled in the art at the time the invention was made to assemble and transmit a message to an electronic means such as a pager, beeper, electronic mail, or smart telephone..." (page 10 of the Office Action). Further, the Examiner cites Bull as providing those tachings. Applicants contend that it would not be obvious to provide the claimed notifying in conjunction with the additionally recited claim features of maintaining the bag buffer, receiving a communication, temporarily storing the input values, searching the bag buffer, and updating variables and/or disposing of input values, which are not taught by either Chess or Bull. Applicants respectfully request reconsideration of the rejections of these claims.

With regard to Claims 20-22, Applicants disagree with the Examiner's conclusion that the claim language is obvious. Applicants first note that the Examiner states that "Chess discloses the data structure of Claim 19." Applicants again remind that Examiner that Claim 19 has been canceled. Further, the rejection has been made without any citation of teachings from the Chess article. Chess simply illustrates, at Figure 2, a sequence of blocks. Chess does not teach or suggest an array data structure, a hash table, or a tuple space data structure, as recited, in the language of Claims 20-22. The Examiner takes Official Notice that data structure are known in the art. Applicants do not traverse the Official Notice that data structures are known. Applicants do, however, traverse the rejection since Chess does not teach or suggest the apparatus as claimed in Claim 18, with or without the known data structures. Applicants respectfully request reconsideration of the rejections of these claims.

Based on the foregoing amendments and remarks, Applicants respectfully request entry of the amendments, reconsideration of the rejections, withdrawal of the rejections, and issuance of the claims.

Respectfully submitted,
Ajay Mohindra, et al

By: Anne Vachon Dougherty

Anne Vachon Dougherty
Registration No. 30,374
Tel. (914) 962-5910